《单片机原理与应用》课程设计报告

专业： 电子信息工程 0

班级： 电信1921 0

姓名： 庞鑫 王婷

学号： 19034117 19034111

商洛学院电子信息与电气工程学院

1. 课程设计目的

随着科技的飞速发展，计算机应用技术日益渗透到社会生产生活的各个领域，而单片机的应用则起到了举足轻重的作用。在工程实践中，经常会遇到各种需要测量转速的场合，例如在发动机、电动机、机床主轴等旋转设备的试验运转和控制中，常需要分时或连续测量、显示其转速及瞬时速度。为了能精确地测量转速，还要保证测量的实时性，要求能测得瞬时转速。因此设计一种较为理想的电机测速控制系统是非常有价值的。本次课程设计是在我们学过单片机后的一次实操，可增加我们的动手能力。 特别是对单片机的系统设计有很大帮助。

二、课程设计使用的设备及开发环境

实验设备：计算机、直流电机模块、 AT89S51单片机、LED数码管

开发环境：keil软件

三、课程设计要求

1、设计分析

本设计分为两部分，**转速检测部分**和**显示部分**。

**转速检测部分**实现了检测直流电机旋转圈数。具体为：直流电机模块输出脉冲传输至AT89S51单片机INTO引脚，并由单片机启动/关闭T0定时器，然后通过圈数和时间之比，计算出直流电机的速度（转/秒）。

**显示部分**实现了显示转速及小组成员学号。具体为：利用P0口和P1口，分别控制段码的发送和位置的选择，P1口选择位置，然后由P0口发送段码并显示，由于人眼的视觉暂留作用，将一组数依次显示，由于最后，将延时时间调整至500us，所以在视觉上是同时显示并刷新的。

2、组成框图

**直流电机模块**

**输出脉冲**

**INT0检测**

**T0定时**

**AT89S51单片机**

**根据时间差计算出转速**

**数码管**

3、设计原理

原理：

直流电机模块光电传感器接受电机扇叶转动的次数获得脉冲，将获得的脉冲传输至AT89S51单片机（由单片机T1口接受高低电平，且计数器在定时器的定时范围内记录脉冲个数）并由单片机计算出直流电机的速度，将经处理后的速度值传输给8段共阴极数码管显示其数值。在此期间学号正常显示，当直流电机的脉冲小于10时，8段共阴极数码管显示一个成员的学号。当直流电机的脉冲大于10时，8段共阴极数码管显示一个成员的学号。

具体：外部中断INT0和定时器T0

1. 直流电机速度显示的实现：

外部中断INT0检测到直流电机转动的脉冲，并打开定时器T0，在等待定时过程中允许溢出中断。直到检测到下一次直流电机转动的脉冲，并重复此过程。AT89S51单片机中计算其转速等于转动的脉冲数与时间的比值。

1. 学号转换的实现：

主函数的执行顺序，当脉冲次数大于10时，8段共阴极数码管显示一个小组成员的学号；当脉冲次数小于10时，8段共阴极数码管显示另一个小组成员的学号。并重复此过程。

四、硬件电路设计

1、VCC：电源电压。

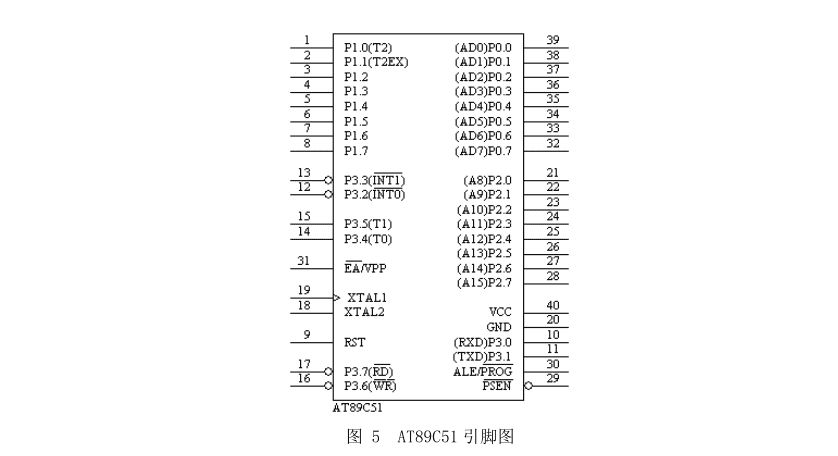
2、GND：地。

3、P1口：P1口是一个8位双向I/O口。口引脚P1.2~P1.7提供内部上拉电阻，P1.0和P1.1要求外部上拉电阻。P1.0和P1.1还分别作为片内精密模拟比较器的同相输入(ANI0)和反相输入(AIN1)。P1口输出缓冲器可吸收20mA电流并能直接驱动LED显示。当P1口引脚写入“1”时，其可用作输入端，当引脚P1.2~P1.7用作输入并被外部拉低时，它们将因内部的写入“1”时，其可用作输入端。当引脚P1.2~P1.7用作输入并被外部拉低时，它们将因内部的上拉电阻而流出电流。4、P3口：P3口的P3.0~P3.5、P3.7是带有内部上拉电阻 的七个双向I/O口引脚。P3.6用于固定输入片内比较器的输出信号并且它作为一通用I/O引脚而不可访问。P3口缓冲器可吸收20mA电流。当P3口写入“1”时，它们被内部上拉电阻拉高并可用作输入端。用作输入时，被外部拉低的P3口脚将用上拉电阻而流出电流。

5、RST：复位输入。RST一旦变成高电平所有的I/O引脚就复位到“1”。当振荡器正在运行时，持续给出RST引脚两个机器周期的高电平便可完成复位。每一个机器周期需12个振荡器或时钟周期。

6、XTAL1：作为振荡器反相器的输入和内部时钟发生器的输入。

7、XTAL2：作为振荡器反相放大器的输出。

**数码管动态显示原理**

动态显示驱动：数码管动态显示接口是单片机中应用最为广泛的一种显示方式之一，动态驱动是将所有数码管的8个显示笔划"a,b,c,d,e,f,g,dp"的同名端连在一起，另外为每个数码管的公共极COM增加位选通控制电路，位选通由各自独立的I/O线控制，当单片机输出字形码时，所有数码管都接收到相同的字形码，但究竟是那个数码管会显示出字形，取决于单片机对位选通COM端电路的控制，所以我们只要将需要显示的数码管的选通控制打开，该位就显示出字形，没有选通的数码管就不会亮。通过分时轮流控制各个数码管的的COM端，就使各个数码管轮流受控显示，这就是动态驱动。在轮流显示过程中，每位数码管的点亮时间为1～2ms，由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位数码管并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感，动态显示的效果和静态显示是一样的，能够节省大量的I/O端口，而且功耗更低。

**霍尔传感器**

霍尔传感器是根据霍尔效应制作的一种磁场传感器。霍尔效应是磁电效应的一种，这一现象是霍尔（A.H.Hall，1855—1938）于1879年在研究金属的导电机构时发现的。后来发现半导体、导电流体等也有这种效应，而半导体的霍尔效应比金属强得多，利用这现象制成的各种霍尔元件，广泛地应用于工业自动化技术、检测技术及信息处理等方面。霍尔效应是研究半导体材料性能的基本方法。通过霍尔效应实验测定的霍尔系数，能够判断半导体材料的导电类型、载流子浓度及载流子迁移率等重要参数。

**直流电动机的工作原理**

在两个珠N、S磁极中间装有一个可以旋转的线圈（abcd），线圈的首尾（a、d）（A、B）相连，电源的“+”、“-”极分别接在与换向片滑动接触的电刷上。当电源接通时，电流通过线圈由“+”→A、a、b、c、d→B“-”构成回路，则线圈的工作边“ab”和“cd”在N、S极磁场作用下，产生两个大小相等方向相反的力，形成了以“O”轴为中心 的逆时针旋转力矩（可利用左手定则判断转子线圈受力方向），于是线圈就按逆时针旋转。当线圈工作边旋转90度至N、S磁极的中性面时，电刷与换向片绝缘物接触，线圈内无电流通过，于是两个工作边不再产生作用力，但由于惯性的作用，线圈仍然旋转。当电与换向片接触时，线圈回路闭合有电流通过，电流从B端进入A端流出。此时，ab工作边位于S极处，cd工作边位于N极处其受力方向与初始情况相同，线圈就这样旋转起来。直流电动机就是根据这个原理制成的

五、软件程序设计

系统程序主要包括主程序、读出速度子程序、速度转换命令子程序、计算速度子程序和显示学号刷新子程序等。

5.1 主程序

主程序的主要功能是负责速度的实时显示、读出并处理直流电机速度，速度测量每1s进行一次。主程序流程图如图4.1所示。 4.2 读出温度子程序

读出速度子程序的主要功能是读出RAM中的9字节。在读出时必须进行CRC校验，校验有错时不能进行速度数据的改写。读出程序框图如下图所示：

**开始**

**数码管的初始化**

**定时器中断初始化**

**定时器定时1ms**

**N**

**继续等待定时计数**

**Y**

**定时器1定时标志位清0**

**外部中断0计时**

**数码管显示**

**结束**

5. 系统调试

系统的调试以程序调试为主。

硬件调试比较简单，首先用万用表测试检查单片机的芯片和电路连接是否能正常接通，然后通过写入相关程序检测数码管能否正常电亮。

软件调试可以先编写显示程序并进行硬件的正确性检验，然后分别进行主程序、读出速度子程序、速度转换命令子程序、计算速度子程序和学号数据刷新子程序等的编程及调试

由于直流电机模块与单片机采用串行数据传送，因此，对直流电机模块进行读/写编程时必须严格地保证读/写时序；否则将无法读取测量结果。本程序采用单片机C语言Keil C51编译器编程调试。

能显示速度值，并且在有速度变化时显示速度能改变，就基本完成。

6. 设计总结

历时2个星期的单片机课程设计已经结束了，在这几个星期的时间里，我们在老师的指导下完成了基于直流电机速度的设计和制作。并在实际操作中发现问题，解决问题，更加增加对知识的认识和理解。

在课程设计的过程中，也遇到了一些问题。比如最开始根据课本上的电路图进行合理的设计布局和布线。我们的布局不合理，导致过程中任务实现相当困难，出现闪烁现象，或根本没有点亮数码管。在之后的程序调试的时候，一点点的解决掉问题，最后出现的成果也能根据直流电机的转速改变，数码管显示的速度也能随之改变。并且不会干扰到学号之间的转变。

1. 参考程序

**/\* sbit a0=P1^7; a1=P1^6;a2=P1^5;a3=P1^4;a4=P1^3; BLANK=P1^2;//BLANK--C[10] S0=P1^1;//decade S1=P1^0;//unit \*/**

**#include "reg51.h" //P0 is the order number of NIXIE LIGHT P1 is the HEX code of NIXIE LIGHT**

**unsigned char f,x;**

**int UNIT,DECADE,v,t;**

**unsigned char code C[]={0X3F,0X06,0X5B,0X4F,0X66,0X6D,0X7D,0X07,0X7F,0X6F,0X40};//HEX code for 0-9**

**//unsigned char code C2[]={0XBF,0X86,0XDB,0XCF,0XE6,0XED,0XFD,0X87,0XFF,0XEF};//HEX code with radix point for 0-9**

**void delay(int i)**

**{**

**int q;**

**for(q=0;q<i;q++)**

**{**

**unsigned char a,b;**

**for(b=71;b>0;b--)**

**for(a=2;a>0;a--);**

**}**

**}**

**void delay500us()**

**{**

**unsigned char a,b;**

**for(b=19;b>0;b--)**

**for(a=5;a>0;a--);**

**}**

**void NiXie\_light(int num)//'num' --- Student ID**

**{**

**unsigned char i;**

**for(i=0;i<8;i++)**

**{**

**switch(i)**

**{**

**case 0: P1=0X7F;**

**//delay(2);**

**P0=C[3];**

**delay500us();**

**P0=0X00;**

**break;**

**case 1: P1=0XBF;**

**//delay(2);**

**P0=C[4];**

**delay500us();**

**P0=0X00;**

**break;**

**case 2: P1=0XDF;**

**//delay(2);**

**P0=C[1];**

**delay500us();**

**P0=0X00;**

**break;**

**case 3: P1=0XEF;**

**//delay(2);**

**P0=C[1];**

**delay500us();**

**P0=0X00;**

**break;**

**case 4: P1=0XF7;**

**//delay(2);**

**P0=C[num];**

**delay500us();**

**P0=0X00;**

**break;**

**case 5: P1=0XFB;**

**//delay(2);**

**P0=C[10];**

**delay500us();**

**P0=0X00;**

**break;**

**case 6: P1=0XFD;**

**//delay(2);**

**P0=C[DECADE];**

**delay500us();**

**P0=0X00;**

**break;**

**case 7: P1=0XFE;**

**//delay(2);**

**P0=C[UNIT];**

**delay500us();**

**P0=0X00;**

**break;**

**}**

**}**

**}**

**void InitTimer0(void)//T0's initialization**

**{**

**TMOD = 0x01;**

**TH0 = 0x0FE;**

**TL0 = 0x0C;**

**EA = 1;**

**ET0 = 1;**

**TR0 = 1;**

**}**

**void InitINT0()//INT0's initialization**

**{**

**IT0=1;**

**EX0=1;**

**EA=1;**

**}**

**void INT0Interrupt() interrupt 0**

**{**

**EX0=0;**

**f++;**

**x++;**

**if(f==7)**

**{**

**TR0=0;**

**v=3500/t;**

**t=0;**

**f=0;**

**ET0=1;**

**TR0=1;**

**}**

**EX0=1;**

**}**

**void Timer0Interrupt(void) interrupt 1 //1MS**

**{**

**TH0 = 0x0FE;**

**TL0 = 0x0C;**

**t++;**

**}**

**void cout()**

**{**

**if(x==40){x=0;}**

**UNIT=(v%10);**

**DECADE=(v/10);**

**}**

**void main()**

**{**

**InitINT0();**

**InitTimer0();**

**while(1)**

**{**

**if(x<20){cout();NiXie\_light(1);}//Change student number every 10 laps**

**if(x>20){cout();NiXie\_light(7);}**

**}**

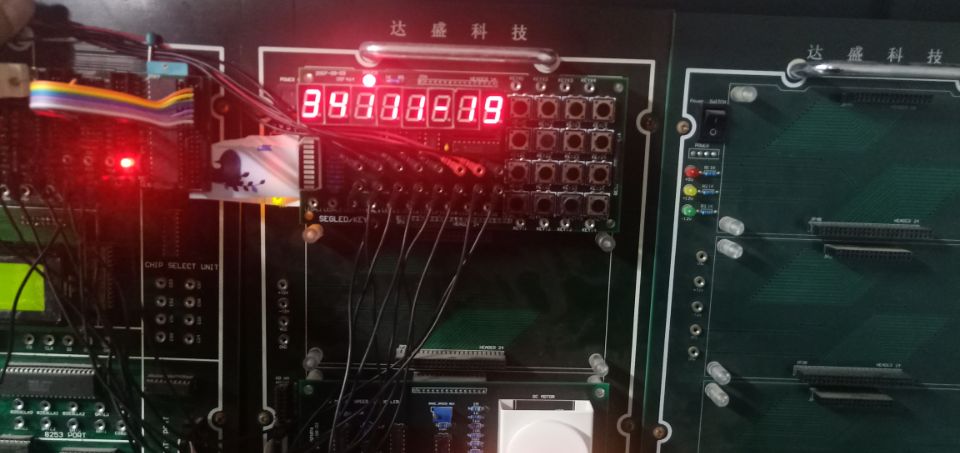
**}**

**六**、**课程设计结果及心得体会**

1、课程设计结果

庞鑫学号（34117）、王婷学号（34111）、显示所测速度（19）





2、课程设计心得体会

刚接触单片机这门课程的时候，我们觉得只要认真听懂课堂上老师讲的理论内容就可以了，可是越到后面的课程就越加觉得只把老师课堂上讲的内容弄明白还是远远不够的。就像这次所做的基于51单片机的直流测速的课程设计，在选到这次的题目的时候我们当时就是懵的状态，在没有任何准备的情况下就要在实验台上进行实操。所幸杨老师专门留出一节课给我们讲了讲51单片机的操作流程以及此次课程设计的设计思路等。经过十六课时的不懈努力，我们总算是将这次的课程设计做出来了。  
 通过本次课程设计，传感器器应用与使用等有了更进一步的认识。同时，也培养了我们的动手能力以及抗压能力，也锻炼了我们自主思考以及团队合作的意识与能力。在本次设计中，我们不但查阅了很多书本上的资料，也通过网络查阅到了很多的知识和芯片的结构接线等方面的知识。

在整个设计中，我们所有程序都是经过不断的编写和修改，并调试硬件才完成。在接下来的子程序编写中，虽然也有不少的困难，但是在大家的努力下问题都得以解诀。